**Détection des spams par SMS grâce à l'apprentissage automatique**

# Introduction

Le service de messagerie SMS, plus connu sous le sigle de SMS (pour « Short Message Service ») ou les noms de « texto » ou de « minimessage », permet de transmettre de courts messages textuels. C'est l'un des services de la téléphonie mobile (il a été introduit par la norme GSM). Parmi ces messages (SMS) certains peuvent avoir un but malveillant, on parle alors de SPAM.

Le spam est un courrier indésirable, aussi appelé pourriel, c’est-à-dire un mail ou un SMS non sollicité, envoyé dans un but publicitaire ou malveillant. Il peut simplement consister en une prospection commerciale qui ne respecterait pas les obligations légales de recueil du consentement des destinataires ou carrément tenter d’arnaquer le destinataire, par des astuces frauduleuses pour gagner de l’argent ou en tentant de le faire cliquer sur des liens de phishing. Il s’agit alors de soutirer des données sensibles à l’utilisateur, ou de diffuser des virus.

De nos jours dans le monde numérique que nous vivons, les données à caractères personnelles représentent l’une des informations cruciales dont la protection est une obligation pour chacun. C’est dans ce contexte que des techniques de classification ont été mise en place pour permettre la détection automatique de ces Spams.

Notre travail consistera de ce fait de mettre en place une de ces techniques de détections des Spam par SMS en utilisant des branches de l’intelligence artificielle à savoir le Machine Learning (l’apprentissage automatique) et à l’aide du traitement du langage naturel (NLP).

Notre travail sera étalé suivant plusieurs étapes. Premièrement nous allons faire une revue littéraire des anciennes techniques utilisées afin de pouvoir ressortir les avantages de l’utilisation de l’apprentissage automatique dans ce contexte. En deuxième étape nous allons mettre en évidence le travail effectué sur les données à partir de la collecte jusqu’au prétraitement. Ensuite nous allons passer à la partie de l’apprentissage automatique combiné au traitement du langage naturel dans laquelle nous allons faire notre choix du model à utiliser. En dernière étape nous allons effectuer le déploiement sur une plateforme de test afin de pouvoir mieux utiliser notre application.

# **Revue littérature**

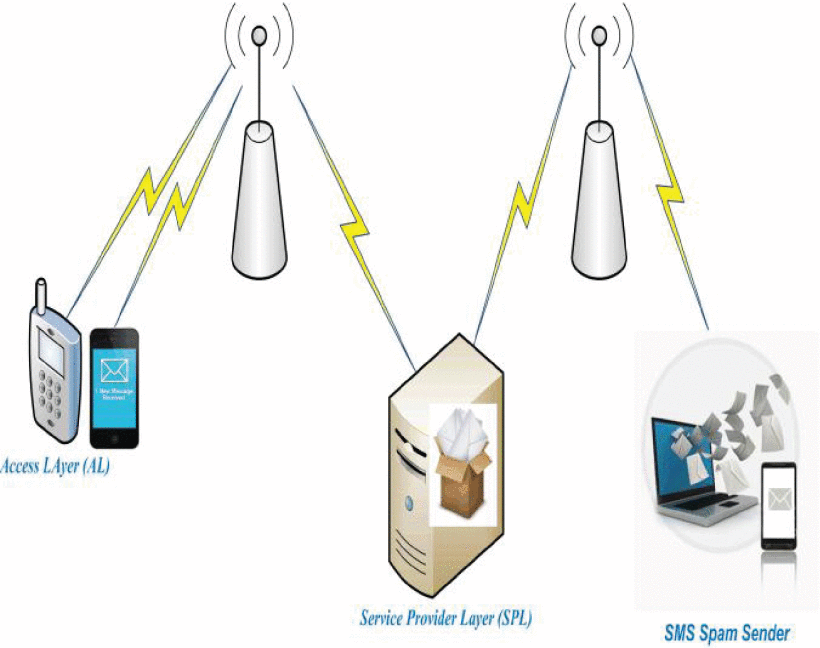
## **Études antérieures sur la détection de spam SMS**

Le spam peut être décrit comme un message électronique non désiré ou non sollicité envoyé en masse à un groupe de destinataires. Les messages sont caractérisés comme étant électroniques, non sollicités, commerciaux, de masse et constituent une menace croissante, principalement en raison des facteurs suivants : 1) la disponibilité de plans de SMS en masse peu coûteux ; 2) la fiabilité (puisque le message atteint l'utilisateur du téléphone mobile) ; 3) la faible probabilité de recevoir des réponses de la part de certains destinataires peu méfiants ; et 4) la possibilité de personnaliser le message. La détection et la prévention du spam par SMS n'est pas une question triviale. Elle a repris un grand nombre de problèmes et de solutions hérités de scénarios relativement anciens de détection et de filtrage du spam par courrier électronique [8]. Les SMS non sollicités sont monnayés courante dans notre vie quotidienne et consomment du temps de communication, de la bande passante et des ressources. Bien que les filtres anti-spam existants offrent un certain niveau de performance, les spams désinforment les destinataires en manipulant des échantillons de données.

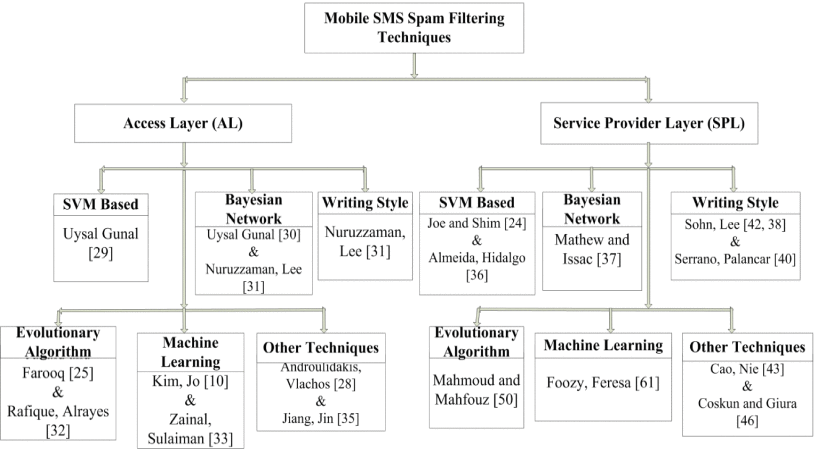
Les études existantes montrent que les techniques de **filtrage** des spams par SMS sont restées à leur stade initial de classification, par exemple la simple similarité des chaînes de caractères ou le blocage explicite des numéros. Les techniques de filtrage traditionnelles telles que le **filtre de classification bayésien**, la régression logistique et l'algorithme de l'arbre de décision pour atténuer les messages de spam SMS prennent encore beaucoup de temps. Des études ont été réalisées sur les différents types de techniques proposées pour le filtrage et l'atténuation du spam SMS mobile. Cependant, il y a encore des progrès à faire en ce qui concerne le filtrage et l'atténuation des spams SMS.

## **Techniques classiques utilisées pour détecter le spam SMS**

Des études récentes sur le spam SMS mobile montrent que plusieurs techniques sont utilisées pour détecter, filtrer ou classer les messages texte de spam. Les solutions sont conçues pour fonctionner soit dans la couche d'accès (AL), soit dans la couche du fournisseur de services (SPL). La couche d'accès est la couche utilisateur, principalement utilisée sous la forme d'un logiciel léger sur la plateforme Android. Un certain nombre de techniques de spamming sont conçues pour fonctionner directement sur le téléphone mobile. D'autres techniques sont conçues pour être déployées au niveau de la SPL. La figure 1 montre l'architecture de base de la ligne de transmission des spams SMS, et la figure 2 présente la taxonomie des recherches sur les spams SMS mobiles.



**Figure 1 : Architecture de la ligne de transmission des spams par SMS.**



**Figure 2 : Taxonomie des techniques de spamming des SMS mobiles.**

### **Access Layer (AL)**

Cette technique a été mise en application par pas mal d’acteurs de la technologie, nous allons prendre l’exemple de **Junaid** et **Farooq** qui ont appliqué un classificateur d'apprentissage évolutionnaire pour créer un système de détection qui filtre les SMS non sollicités au niveau de la couche d'accès d'un téléphone mobile. Ils étudient un message SMS dans le système hexadécimal et extraient deux caractéristiques de ce format, les bigrammes octet1 et la distribution de fréquence des octets. Ils évaluent la praticabilité d'un certain nombre de classificateurs évolutifs et non évolutifs (fonctionnels sur les ensembles de caractéristiques excédentaires) pour le système de filtrage. Les résultats des expériences suggèrent que le système de classification supervisé (SCS) fonctionnant sur l'ensemble de caractéristiques atteint un taux de détection supérieur à 89 % et un taux de fausse alarme proche de 0 %.

Nous avons aussi Kim et al qui ont mis au point un ratio de fréquence des mots clés et un schéma de simulation de l'outil d'apprentissage automatique WEKA 3.7 pour le système léger et rapide. Le filtre de messages mobiles peut être exécuté de manière autonome dans les téléphones grâce à l'utilisation du rapport de fréquence des mots-clés (FR). Chaque message est décomposé en un ensemble de mots-clés en utilisant la fonction "string to word vector" dans une interface WEKA. Le prétraitement est ensuite effectué sur 5 574 messages SMS.

### **Service Providers Layer (SPL)**

Joe et Shim [24] ont utilisé un SVM pour filtrer le spam dans le système mobile en appliquant un apprentissage basé sur l'expérience pour identifier les SMS non sollicités. Les termes contenus dans le texte d'un SMS sont extraits en passant par un préprocesseur et un dictionnaire. Si le terme homogénéisé est contenu dans la liste des caractéristiques, le catalogue de mots est fixé à 1 ou 0. Les valeurs vectorielles produites sont utilisées comme données d'apprentissage pour modifier l'hyperplan du SVM. Une fois que chaque vecteur de caractéristiques est marqué 0 ou 1, un processus d'apprentissage est conclu à partir du classificateur SVM latéral. Une fonction de base radiale (RBF) gaussienne est utilisée comme fonction noyau. La valeur constante est fixée à 10, 20, 40, et les valeurs gamma sont fixées à 0,01, 0,05 et 0,1. La technique montre ses performances avec un taux de vecteur de caractéristiques de 150, un taux constant de 20 et un taux de gamma de 0,01. Le taux de détection est considérablement réduit lorsque le dispositif de prétraitement ne peut pas séparer correctement les lignes de mots.

Mahmoud et Mahfouz [50] ont créé un système immunitaire artificiel (SIA) de classification des SMS pour filtrer le spam par SMS. Le système AIS utilise un ensemble de caractéristiques pour servir de filtre anti-spam en entrée. Il catégorise les messages textuels à l'aide d'un ensemble de données formées qui se compose de numéros de téléphone, de mots de spam et de détecteurs. Les résultats expérimentaux sont obtenus en utilisant le système d'exploitation iPhone (iOS). Les résultats de cette expérience montrent que le système proposé est capable de classer les messages en tant que spam ou non-spam avec plus de précision et de vitesse de convergence que l'algorithme bayésien naïf.

## **Avantages de l'utilisation de l'apprentissage automatique dans ce contexte**

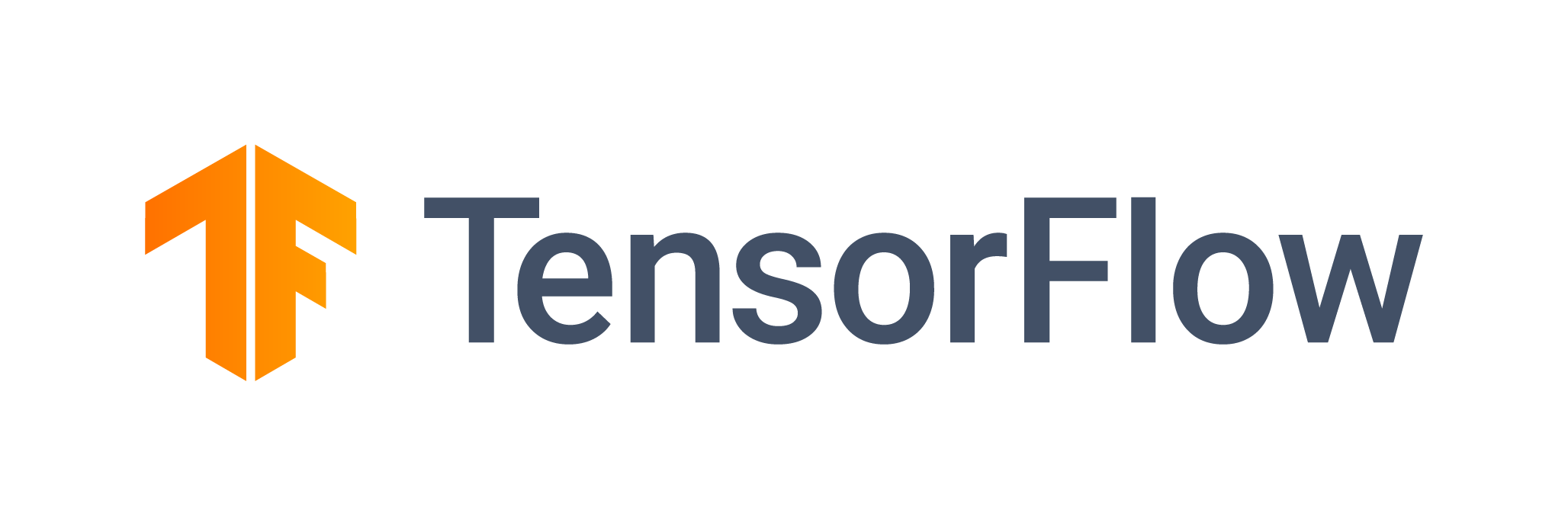
L'apprentissage automatique est une branche de l'intelligence artificielle (IA) et de l'informatique qui porte sur l'utilisation des données et des algorithmes pour imiter la manière dont les êtres humains apprennent, afin d'améliorer progressivement sa précision.

Au cours des deux dernières décennies, les avancées technologiques dans le domaine du stockage et de la puissance de traitement ont permis de créer des produits innovants basés sur l'apprentissage automatique, tels que le moteur de recommandation de Netflix et les voitures autonomes.

L'apprentissage automatique est une composante importante du domaine en pleine expansion qu'est la science des données. Grâce à l'utilisation de méthodes statistiques, des algorithmes sont entraînés à effectuer des classifications ou des prévisions, ce qui permet de découvrir des informations essentielles dans le cadre de projets d'exploration de données. Ces informations permettent ensuite de prendre des décisions dans les applications et les entreprises, et ont idéalement un impact sur les principales mesures de croissance. Avec l'essor du Big Data, la demande de spécialistes des données va continuer d'augmenter. Ces derniers devront identifier les questions économiques les plus pertinentes et les données permettant d'y répondre.

Les algorithmes d'apprentissage automatique sont généralement créés à l'aide d'infrastructures qui accélèrent le développement de solutions, tels que TensorFlow et PyTorch.

**TensorFlow** est un outil open source d'apprentissage automatique développé par Google.



**Figure 3 : TensorFlow**

**PyTorch** est une bibliothèque logicielle Python open source d'apprentissage machine qui s'appuie sur Torch développée par Meta.



**Figure 4 : PyTorch**

L'utilisation de l'apprentissage automatique pour la détection de spam SMS présente plusieurs avantages, notamment une meilleure capacité d'adaptation aux changements dans les modèles de spam, une précision accrue et une réduction du besoin de règles manuelles. Voici quelques-uns des avantages clés :

* **Adaptabilité aux Modèles de Spam Evolutifs :**

Les modèles de spam SMS évoluent constamment. L'apprentissage automatique peut s'adapter dynamiquement aux nouveaux schémas de spam sans nécessiter une mise à jour manuelle constante des règles.

* **Capacité à Gérer des Grandes Quantités de Données :**

L'apprentissage automatique est particulièrement efficace lorsqu'il est confronté à de grandes quantités de données. Les modèles peuvent être entraînés sur d'énormes ensembles de données, ce qui peut améliorer la capacité du modèle à généraliser et à détecter des schémas subtils.

* **Réduction des Faux Positifs :**

En utilisant des modèles d'apprentissage automatique, la détection de spam peut être plus précise, réduisant ainsi le nombre de faux positifs (messages légitimes marqués comme spam). Cela améliore l'expérience de l'utilisateur en évitant la perte de messages importants.

* **Apprentissage à Partir de l'Expérience :**

Les modèles d'apprentissage automatique peuvent apprendre à partir de l'expérience en ajustant leurs poids et leurs paramètres en fonction des erreurs passées. Cela permet une amélioration continue de la précision du modèle au fil du temps.

* **Flexibilité dans le Choix des Caractéristiques :**

Les modèles d'apprentissage automatique offrent la flexibilité de choisir parmi diverses caractéristiques, telles que la fréquence des mots, la longueur du message, la présence de liens, etc., pour améliorer la détection du spam.

* **Réaction Rapide aux Changements :**

L'apprentissage automatique permet une réaction rapide aux changements dans les tendances du spam. Les modèles peuvent être ré-entraînés régulièrement pour rester efficaces dans la détection des schémas de spam émergents.

En vue de ces précédents avantages cités sur l’utilisation de l’apprentissage automatique pour la détection des Spam SMS, nous allons maintenant dans la partie suivante faire une étude détaillée de cette méthode avec un cas pratique à l’appui.

# **Collecte et Analyse des données**

Dans tout processus d’apprentissage automatique les données représentent un facteur primordial. Sans données, aucune opération ne pourrait être effectuée.

Dans cette partie nous allons faire une étude détaillée de nos données allant de la collecte, à l’exploration jusqu’au prétraitement.

## **Collecte des données**

### **Source de données :**

Dans notre cas pratique nous avons utilisé un jeu de données public du nom de **« SMS Spam Collection ».** La collection de SMS Spam est un ensemble de messages SMS étiquetés qui ont été collectés pour la recherche sur les SMS Spam. Elle contient un ensemble de SMS en anglais de 5 572 messages, étiquetés selon deux catégories à savoir "ham" (légitimes) ou "spam".

Les fichiers contiennent un message par ligne. Chaque ligne est composée de deux colonnes : v1 contient l'étiquette (ham ou spam) et v2 contient le texte brut.

Ce corpus a été collecté à partir de sources gratuites ou libres pour la recherche sur Internet :

Une collection de 425 messages de spam SMS a été extraite manuellement du site Web Grumbletext. Il s'agit d'un forum britannique dans lequel les utilisateurs de téléphones portables font des déclarations publiques sur les messages de spam SMS, la plupart du temps sans signaler le message de spam reçu. L'identification du texte des messages de spam dans les réclamations est une tâche très difficile et fastidieuse, qui a nécessité l'analyse minutieuse de centaines de pages web. Le site web de Grumbletext est le suivant [Lien Web].

Un sous-ensemble de 3 375 SMS choisis au hasard parmi les messages de spam du NUS SMS Corpus (NSC), qui est un ensemble de données d'environ 10 000 messages légitimes collectés à des fins de recherche au département d'informatique de l'université nationale de Singapour. Les messages proviennent en grande partie de Singapouriens et surtout d'étudiants de l'université. Ces messages ont été collectés auprès de volontaires qui ont été informés que leurs contributions allaient être rendues publiques. Le corpus SMS de la NUS est disponible à l'adresse suivante : [Lien web].

Une liste de 450 messages SMS ham recueillis dans le cadre de la thèse de doctorat de Caroline Tag est disponible à l'adresse suivante : [Web Link].

Enfin, nous avons intégré le SMS Spam Corpus v.0.1 Big. Il contient 1 002 SMS ham et 322 messages de spam et est public.

Voici un aperçu du jeu de données en question :

|  | **label** | **message** |
| --- | --- | --- |
| 0 | ham | Go until jurong point, crazy.. Available only ... |
| 1 | ham | Ok lar... Joking wif u oni... |
| 2 | spam | Free entry in 2 a wkly comp to win FA Cup fina... |
| 3 | ham | U dun say so early hor... U c already then say... |
| 4 | ham | Nah I don't think he goes to usf, he lives aro... |
| ... | ... | ... |
| 5567 | spam | This is the 2nd time we have tried 2 contact u... |
| 5568 | ham | Will ü b going to esplanade fr home? |
| 5569 | ham | Pity, \* was in mood for that. So...any other s... |
| 5570 | ham | The guy did some bitching but I acted like i'd... |
| 5571 | ham | Rofl. Its true to its name |

**Tableau 1 : Jeu de données SMS Spam**

Une fois le jeu de données choisit, nous aurons ensuite besoin de le charger dans notre IDE afin de pouvoir faire les manipulations nécessaires.

Pour ce faire nous avons fait recours à une bibliothèque python du nom de **Pandas.**

**Pandas** est une bibliothèque écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles.

Pandas est un logiciel libre sous licence BSD2. Son nom est dérivé du terme Panel Data (en français "données de panel", un terme d'économétrie pour les jeux de données qui comprennent des observations sur plusieurs périodes de temps pour les mêmes individus). Son nom est également un jeu de mots sur l'expression "Python Data Analysis".



**Figure 5 : Pandas**

## **Exploration des données**

L'exploration de données est une technique assistée par ordinateur utilisée en analyse pour traiter et explorer de grands ensembles de données. Grâce aux outils et méthodes d'exploration de données, les organisations peuvent découvrir des tendances et des relations cachées dans leurs données. L'exploration des données transforme les données brutes en connaissances pratiques. Les entreprises utilisent ces connaissances pour résoudre des problèmes, analyser l'impact futur des décisions commerciales, et augmenter leurs marges bénéficiaires.

L'extraction de règles d'association est le processus qui consiste à trouver des relations entre deux ensembles de données différents, apparemment sans rapport. Les énoncés « si-alors » démontrent la probabilité d'une relation entre deux points de données. Les scientifiques des données mesurent la précision des résultats à l'aide de critères de soutien et de confiance. La prise en charge mesure la fréquence d'apparition des éléments liés dans l'ensemble de données, tandis que la confiance indique le nombre de fois où une déclaration de type "si-alors" est exacte.

### **Classification**

La classification est une technique complexe d'exploration de données qui entraîne l'algorithme ML à trier les données en catégories distinctes. Elle utilise des méthodes statistiques telles que les arbres de décision et les plus proches voisins pour identifier la catégorie. Dans toutes ces méthodes, l'algorithme est préprogrammé avec des classifications de données connues pour deviner le type d'un nouvel élément de données.

Par exemple, les analystes peuvent entraîner le logiciel d'exploration de données en utilisant des images étiquetées de pommes et de mangues. Avec une certaine précision, le logiciel peut alors prédire si une nouvelle image est une pomme, une mangue ou un autre fruit.

### **Clustering**

Le clustering consiste à regrouper plusieurs points de données en fonction de leurs similitudes. Il est différent de la classification car il ne peut pas distinguer les données par catégorie spécifique mais peut trouver des modèles dans leurs similarités. Le résultat de l'exploration de données est un ensemble de clusters où chaque collection est distincte des autres groupes, mais où les objets de chaque cluster sont similaires d'une certaine manière.

Par exemple, l'analyse cluster peut être utile pour les études de marché lorsqu'on travaille avec des données multivariées provenant d'enquêtes. Les spécialistes des études de marché utilisent l'analyse cluster pour diviser les consommateurs en segments de marché et mieux comprendre les relations entre les différents groupes.

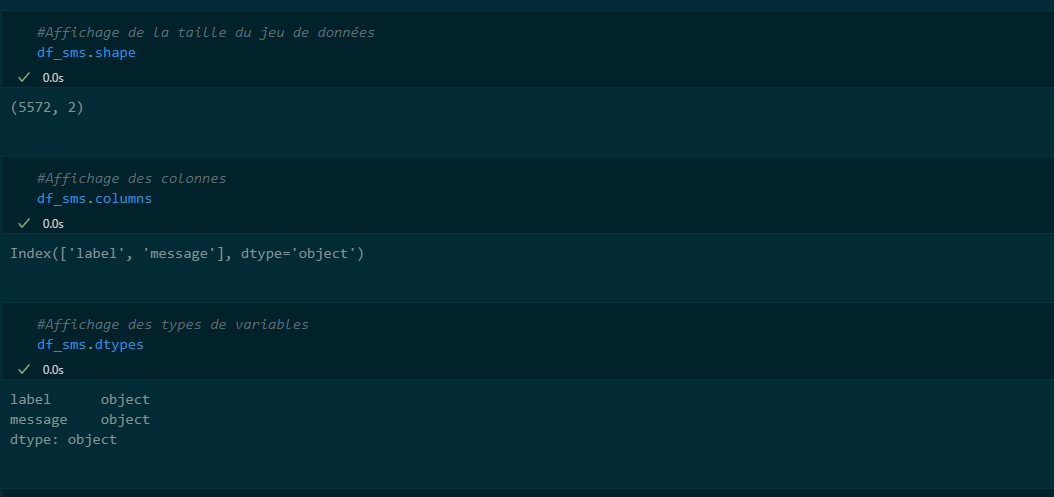
Dans cette partie nous allons procéder à l’exploration de notre jeu de donnée.

### **Description des caractéristiques des données**

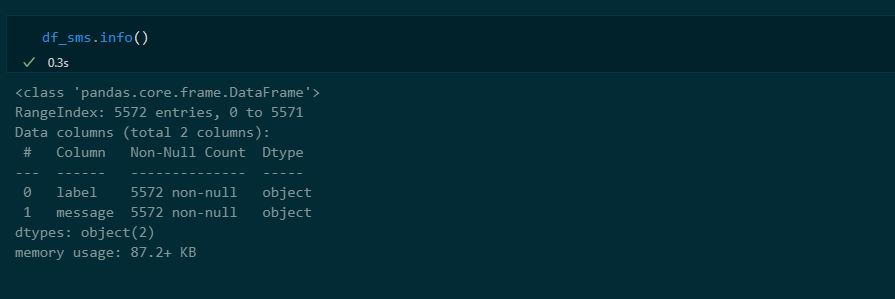
Comme mentionné précédemment le jeu de données est composé de deux colonnes et de **5572** lignes. Les variables sont de types **Object** séparées en deux colonnes : **label** qui représente le type de message (hpam ou spam) et **message** qui contient le texte.

Le jeu de données ne présente ni de valeurs nulles ni de valeurs dupliquées.

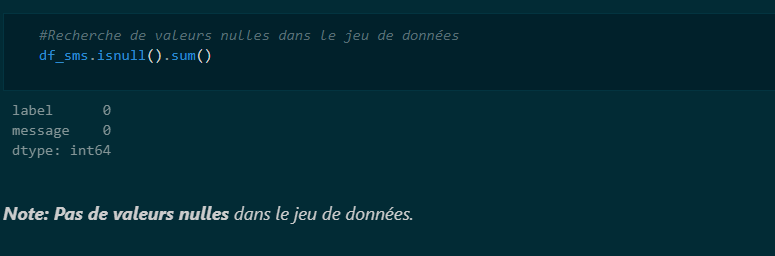
Le processus de recueille de ses informations s’est fait comme suit :



**Capture 1 : Affichage des détails du jeu de données**

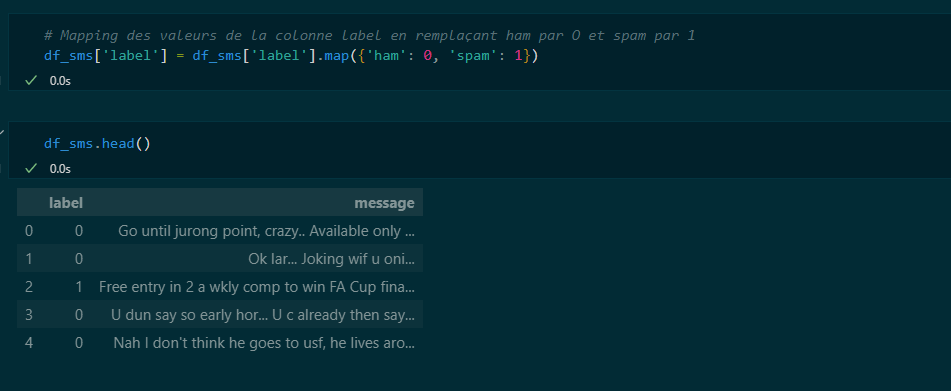


**Capture 1 : Affichage des infos du jeu de données**



**Capture 3 : Vérification de l’existence de valeurs nulles**

Pour une manipulation plus facile de nos données nous avons effectué un **Mapping** au niveau de la colonne **Label** et changer les valeurs ham en 0 et Spam en 1.



**Capture 4 : Mapping de la colonne Label**

### **Statistiques descriptives**

Nous avons ensuite affiché les statistiques descriptives du jeu de données :

|  | **label** | **message** |
| --- | --- | --- |
| count | 5572 | 5572 |
| unique | 2 | 5169 |
| top | ham | Sorry, I'll call later |
| freq | 4825 | 30 |

Tableau 2 : Statistiques descriptives du jeu de données

Interprétation :

**Label** : count (nombre) : Il y a un total de 5572 observations (messages) dans l’ensemble de données.

Unique (unique) : Il y a 2 catégories uniques dans la colonne "label". Cela signifie qu'il y a deux labels possibles. Dans ce cas, les labels sont "ham" (non-spam) et Spam.

top (plus fréquent) : "ham" est la catégorie la plus fréquemment observée dans la colonne "label". Il y a 4825 occurrences de "ham".

freq (fréquence) : La fréquence correspondante à "ham" est de 4825. Cela signifie que "ham" est l'étiquette la plus fréquemment observée dans l'ensemble de données.

**Message** : count (nombre) : Il y a également 5572 observations (messages) dans la colonne "message". Chaque message est associé à un label dans la colonne "label".

Unique (unique) : Il y a 5169 messages uniques dans la colonne "message".

Top (plus fréquent) : "Sorry, I'll call later" est le message le plus fréquemment observé dans l'ensemble de données.

Freq (fréquence) : La fréquence correspondante à ce message est de 30. Cela signifie que ce message particulier apparaît 30 fois dans l'ensemble de données.

### **Graphe de visualisation de la distribution des classes**

Avec le développement des technologies intelligentes qui génèrent des quantités astronomiques de data, la visualisation des données devient indispensable. En effet, pour pouvoir analyse ses Big Data et les utiliser au mieux dans sa stratégie d’entreprise, il est essentiel de pouvoir les lire et répertorier ses informations business dans des tableaux de bord visuels.

La visualisation des données (ou data visualization ou dataviz en anglais) désigne le fait de représenter visuellement ses data pour pouvoir déceler et comprendre des informations, les données brutes étant difficilement interprétables et exploitables. Ce processus se fait par des outils analytiques spécifiques et se matérialise par des tableaux (type Excel), des graphiques, des cartes visuelles ou même des infographies regroupées dans des Dashboard (tableaux de bord).

En classant, segmentant et scénarisant les données visuellement, une entreprise peut déceler des informations auparavant inaccessibles en un simple coup d’œil. La visualisation des données permet donc à toute organisation de piloter son activité plus efficacement en adoptant une stratégie data-driven et agile.

Dans le domaine de la data science plus précisément avec **Python** nous avons utilisées deux principales librairies pour effectuer la visualisation de la distribution de nos données.

La première librairie est **matplotlib.** Matplotlib est une bibliothèque Python open source, initialement développée par le neurobiologiste John Hunter en 2002. L’objectif était de visualiser les signaux électriques du cerveau de personnes épileptiques. Pour y parvenir, il souhaitait répliquer les fonctionnalités de création graphique de MATLAB avec Python.

Suite au décès de John Hunter en 2012, Matplotlib a été améliorée au fil du temps par de nombreux contributeurs de la communauté open source. Elle est utilisée pour créer des graphiques, et des diagrammes de haute qualité. C’est une alternative open source à MATLAB.

Il est par exemple possible de créer des tracés, des histogrammes, des diagrammes à barre et tous types de graphiques à l’aide de quelques lignes de code. Il s’agit d’un outil très complet, permettant de générer des visualisations de données très détaillées.

Cette bibliothèque est particulièrement utile pour les personnes travaillant avec Python ou NumPy. Elle est notamment utilisée sur des serveurs d’application web, des shells et des scripts Python. Avec les APIs de matplotlib, il est aussi possible pour les développeurs d’intégrer des graphiques à des applications d’interface graphique.

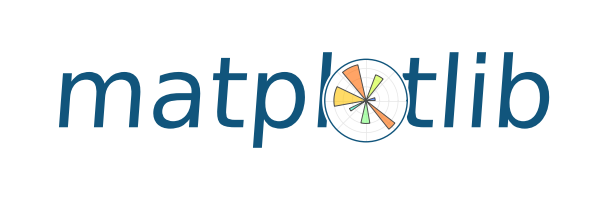


Figure 6 : Matplotlib

La deuxième bibliothèque est **seaborn**. La librairie **Seaborn** vient proposer une alternative à Matplotlib. C’est également une librairie permettant de générer des graphiques, tout comme Matplotlib. Seaborn est exactement pareil vis-à-vis de Matplotlib. La librairie reprend les mêmes principes que Matplotlib, mais en ajoutant quelque chose “par-dessus”, afin d’offrir des qualités de graphiques différents, des nouvelles fonctionnalités, etc. Si vous me permettez une dernière image, on pourrait imaginer que Seaborn est une voiture flambant neuve dont l’ensemble du châssis et le moteur seraient Matplotlib.

Seaborn vient corriger plusieurs défauts de Matplotlib :

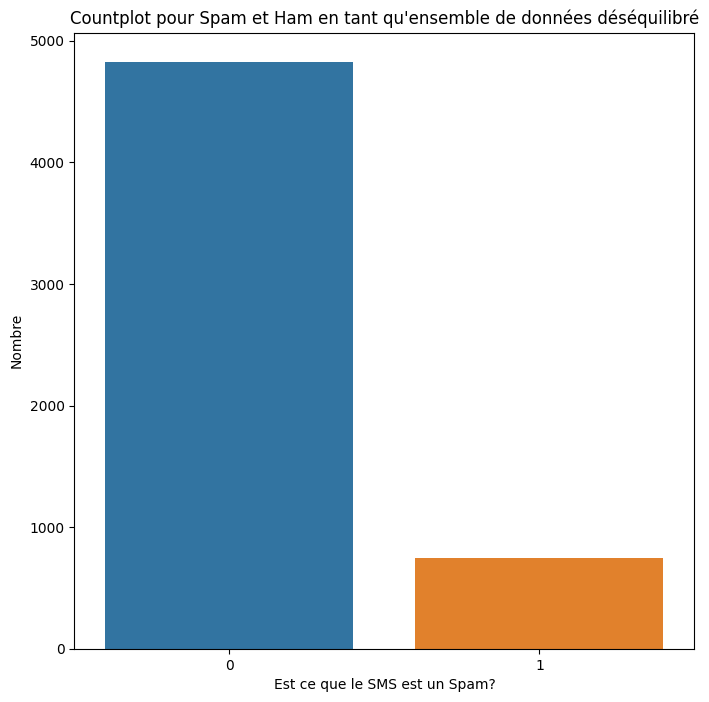
* Il propose de multiples modèles graphiques prédéfinis de bonne qualité esthétique, en modifiant les options graphiques par défaut de Matplotlib ;
* Il ajoute une interaction avec les data frames afin de faciliter grandement la génération de graphiques à partir de ceux-ci ;
* Il propose un catalogue – très – dense de fonctions graphiques pour répondre le plus précisément possible à une problématique donnée.

En contrepartie, Seaborn propose un peu moins d’options de personnalisation en tant que telles.



Figure 7 : Seaborn

L’utilisation de ces deux bibliothèques nous a permis de pouvoir visualiser la distribution de nos données comme le montre la figure ci-après :



**Figure 8 : Graphique de la distribution des données**

En se basant sur ce graphique nous pouvons observer la distribution des données qui est totalement déséquilibré. Nous notons une grande différence entre le nombre de message Ham et les messages SPAM.

Nous allons de ce fait procéder à l’équilibrage du jeu de données afin de bien mener le traitement.

Pour cela nous allons faire recours à certaines techniques comme le **Oversampling** et le **Feature Engineering.**

### **Oversampling**

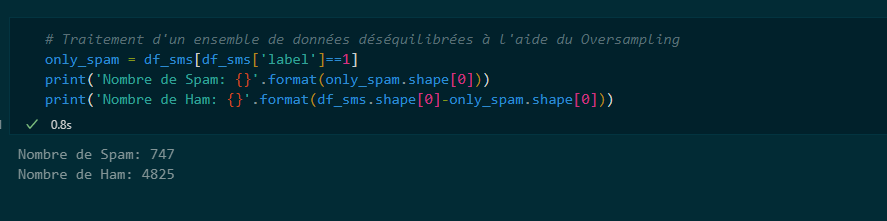
Le suréchantillonnage peut être défini comme le processus consistant à augmenter artificiellement le nombre d'instances dans la classe minoritaire en créant des points de données synthétiques. Ces nouveaux points de données sont générés sur la base des caractéristiques des instances existantes de la classe minoritaire. En introduisant ces instances synthétiques, le suréchantillonnage garantit que les deux classes ont une représentation relativement égale dans l'ensemble de données.

Voyons plus en détail comment fonctionne le suréchantillonnage. Lorsqu'il s'agit d'ensembles de données déséquilibrés, la classe minoritaire a souvent moins d'instances que la classe majoritaire. Ce déséquilibre entre les classes peut conduire à des modèles biaisés qui peinent à prédire avec précision la classe minoritaire. Pour surmonter ce problème, les techniques de suréchantillonnage entrent en jeu.

Une technique de suréchantillonnage couramment utilisée est la technique de suréchantillonnage synthétique des minorités (Synthetic Minority Over-sampling Technique, SMOTE). La technique SMOTE consiste à créer des instances synthétiques le long des segments de ligne reliant les instances de la classe minoritaire. Cette approche garantit que les instances synthétiques se situent dans l'espace des caractéristiques de la classe minoritaire, ce qui les rend représentatives de la distribution sous-jacente.

L'algorithme ADASYN (Adaptive Synthetic Sampling) est une autre technique de suréchantillonnage. ADASYN se concentre sur la génération d'instances synthétiques pour la classe minoritaire qui sont plus difficiles à classer. Ce faisant, ADASYN vise à résoudre le problème du chevauchement des régions entre la classe minoritaire et la classe majoritaire, améliorant ainsi la capacité du modèle à les distinguer.

Afin de procéder à l’équilibrage de la distribution de nos données nous avons opté de faire un Oversampling comme le montre la capture de notre code ci-après :



Explication du code :

Ligne 1 : only\_spam = df\_sms[df\_sms['label']==1]

Cette ligne de code crée un nouveau DataFrame appelé only\_spam qui contient uniquement les lignes où la colonne 'label' a la valeur 1, ce qui indique que le message est spam. En d'autres termes, cela filtre le DataFrame original (df\_sms) pour inclure uniquement les exemples de la classe spam.

Ligne 2 : print('Nombre de Spam: {}'.format(only\_spam.shape[0]))

Cette ligne affiche le nombre de messages spam dans l'ensemble de données en utilisant la fonction shape[0] qui renvoie le nombre de lignes dans le DataFrame only\_spam. La chaîne formatée est utilisée pour inclure cette information dans le message imprimé.

Ligne 3 : print('Nombre de Ham: {}'.format(df\_sms.shape[0]-only\_spam.shape[0]))

Cette ligne affiche le nombre de messages non-spam (ham) dans l'ensemble de données. Elle utilise la propriété shape[0] du DataFrame df\_sms pour obtenir le nombre total de messages, puis soustrait le nombre de messages spam (only\_spam.shape[0]) pour obtenir le nombre de messages ham.

En résumé nous avons effectué une analyse rapide du nombre de message spam et ham.

Ensuite nous avons effectué le suréchantillonnage de la classe minoritaire spam.

count = int((df\_sms.shape[0]-only\_spam.shape[0])/only\_spam.shape[0])

for i in range(0, count-1):

  df\_sms = pd.concat([df\_sms, only\_spam])

df\_sms.shape

Ligne 1 : count = int((df\_sms.shape[0]-only\_spam.shape[0])/only\_spam.shape[0])

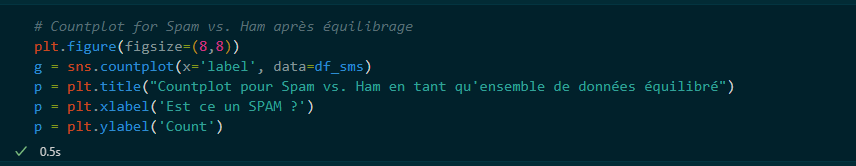
Sur cette ligne nous avons calculé le nombre de fois que les exemples de la classe minoritaure spam doivent être dupliqués pour équilibrer la classe.

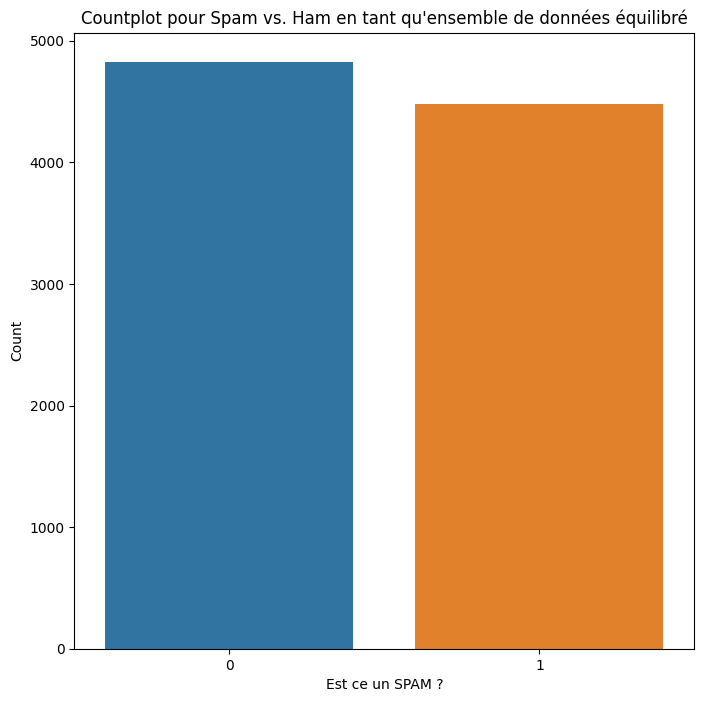
Ligne 2 : for i in range(0, count-1):

  df\_sms = pd.concat([df\_sms, only\_spam])

Nous avons ensuite utilisé une boucle for et à l’aide de la librairie **pandas**, faire une concaténation des jeux de données *df\_sms* et *only\_spam* autant de fois que la valeur calculé dans la ligne précédente.

Afin de pouvoir visualiser les modifications effectuées grâce au suréchantillonnage, affichons le graphique des nouvelles données.





**Figure 8 : Graphique de la distribution des données après suréchantillonnage**

### **Feature Engineering**

Le Feature Engineering est un processus qui consiste à transformer des données brutes en caractéristiques qui représentent plus précisément le problème sous-jacent pour un modèle prédictif

En d'autres termes, il s'agit d'appliquer les connaissances du domaine pour extraire des représentations analytiques des données brutes et les préparer pour l'apprentissage automatique.

Il s'agit de la première étape du développement d'un modèle prédictif d'apprentissage automatique. Elle permet d'accroître la précision du modèle sur des données nouvelles et inédites.

Il est important de se rappeler que les algorithmes d'apprentissage automatique apprennent une solution à un problème à partir d'un échantillon de données. Le Feature Engineering détermine donc la meilleure représentation de l'échantillon de données pour l'apprentissage de la solution au problème.

Ceci est très important car le succès d'un projet d'intelligence artificielle ou d'apprentissage automatique dépend souvent de la représentation des données. Les algorithmes doivent être en mesure de comprendre les données d'entrée.

Le Feature Engineering repose sur un ensemble de procédures et de méthodes bien définies. Les procédures à utiliser varient en fonction des données, et c'est par l'expérience et la pratique que l'on apprend lesquelles utiliser dans un contexte donné.

Voici quelques aspects clés du feature engineering :

**Sélection des Caractéristiques** : Identifiez les caractéristiques les plus importantes pour le problème que vous essayez de résoudre. La sélection judicieuse des caractéristiques peut améliorer l'efficacité du modèle et réduire la complexité.

**Création de Nouvelles Caractéristiques** : Générez de nouvelles caractéristiques à partir des caractéristiques existantes qui peuvent capturer des relations non linéaires ou des informations importantes. Par exemple, créer une caractéristique "ratio" ou "somme" à partir de deux caractéristiques existantes.

**Transformation des Données** : Appliquez des transformations sur les caractéristiques existantes pour les rendre plus adaptées au modèle. Cela peut inclure des opérations comme la normalisation, la standardisation, ou l'application de fonctions mathématiques.

**Gestion des Valeurs Manquantes** : Développez des stratégies pour traiter les valeurs manquantes dans les données. Cela peut impliquer l'imputation (remplacement) des valeurs manquantes par des moyennes, des médianes, ou d'autres stratégies basées sur le contexte.

**Encodage des Catégories** : Convertissez les variables catégorielles en une forme que les modèles d'apprentissage automatique peuvent comprendre. Cela peut inclure l'encodage one-hot, l'encodage ordinal, ou d'autres techniques d'encodage.

Dans notre cas nous allons utiliser l’aspect de **création de nouvelles caractéristiques.**

Nous avons créé une nouvelle colonne nommé **word\_count :**

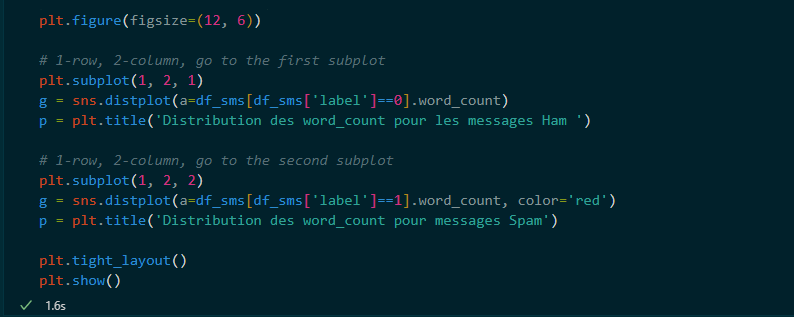
df\_sms['word\_count'] = df\_sms['message'].apply(**lambda** x: len(x.split()))

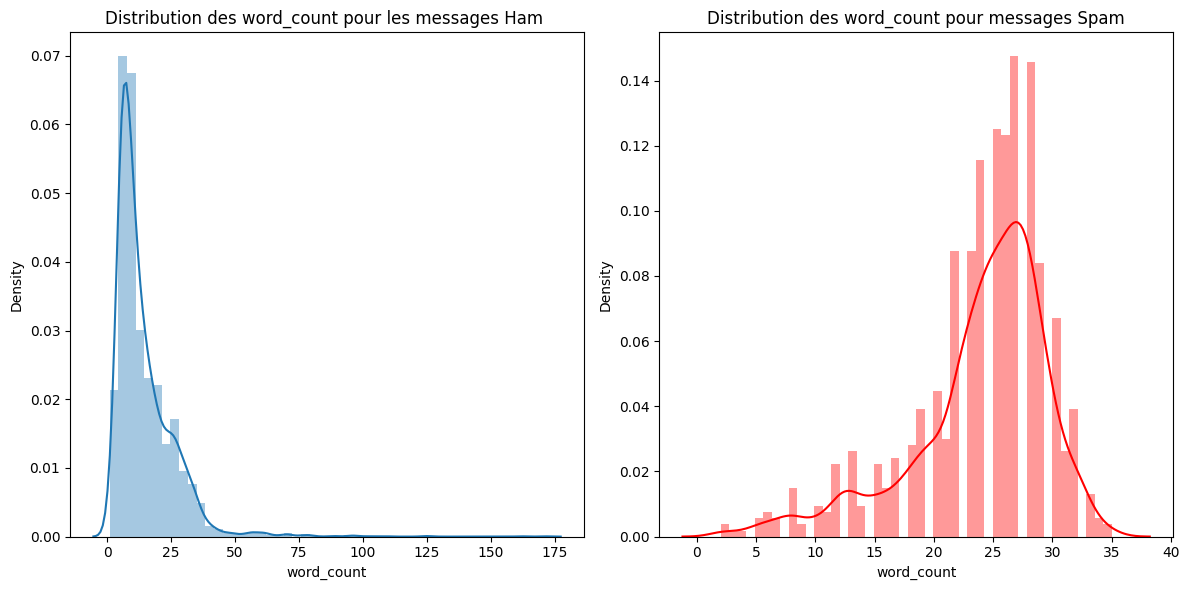
Cette colonne va enregistrer le nombre de mots dans chaque message, comme le montre le jeu de données ci-dessous :

|  | **label** | **message** | **word\_count** |
| --- | --- | --- | --- |
| 0 | 0 | Go until jurong point, crazy.. Available only ... | 20 |
| 1 | 0 | Ok lar... Joking wif u oni... | 6 |
| 2 | 1 | Free entry in 2 a wkly comp to win FA Cup fina... | 28 |
| 3 | 0 | U dun say so early hor... U c already then say... | 11 |
| 4 | 0 | Nah I don't think he goes to usf, he lives aro... | 13 |

Tableau 3 : Nouveau dataset avec l’ajout de la colonne word\_count

Nous avons ensuite fait l’affichage des graphiques pour visualiser la distribution des word\_count pour chaque type de message.

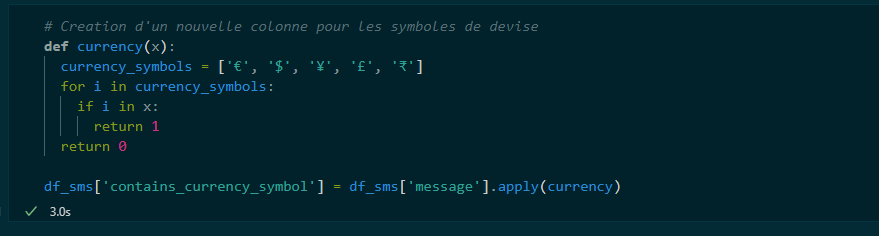




**Figure 9 : Graphique de la distribution des word\_count pour les chaque type de message**

On remarque le nombre de mots des messages de spam se situe entre 15 et 30 mots, alors que la majorité des messages de spam se situent en dessous de 25 mots.

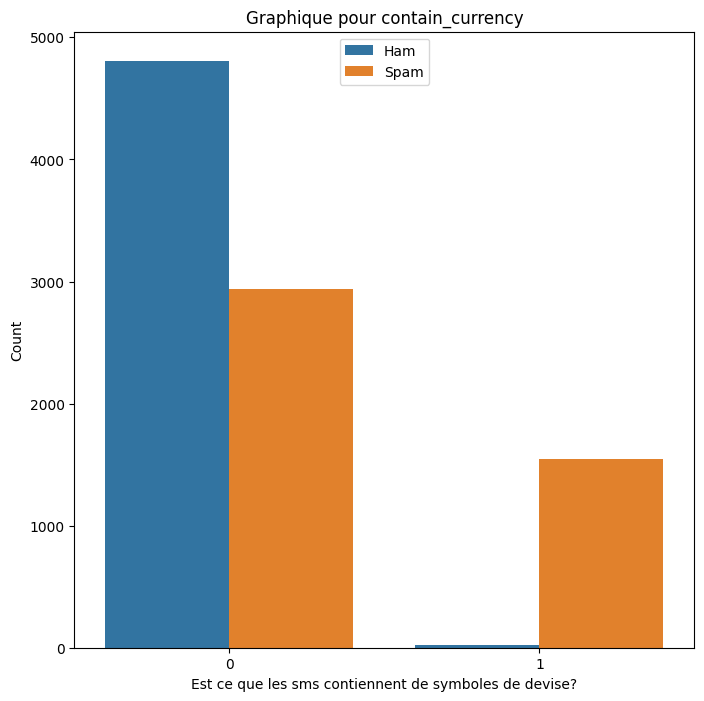
Nous avons ensuite procédé à la création d’une autre colonne *contains\_currency\_symbol* pour répertorier les symboles de devise présentent dans les messages.



|  | **label** | **message** | **word\_count** | **contains\_currency\_symbol** |
| --- | --- | --- | --- | --- |
| 5537 | 1 | Want explicit SEX in 30 secs? Ring 02073162414... | 16 | 0 |
| 5540 | 1 | ASKED 3MOBILE IF 0870 CHATLINES INCLU IN FREE ... | 33 | 1 |
| 5547 | 1 | Had your contract mobile 11 Mnths? Latest Moto... | 28 | 0 |
| 5566 | 1 | REMINDER FROM O2: To get 2.50 pounds free call... | 28 | 0 |
| 5567 | 1 | This is the 2nd time we have tried 2 contact u... | 30 | 1 |

En affichant de nouveau jeux de données avec la nouvelle colonne on peut voir la présence de symboles de devise dans certains messages.

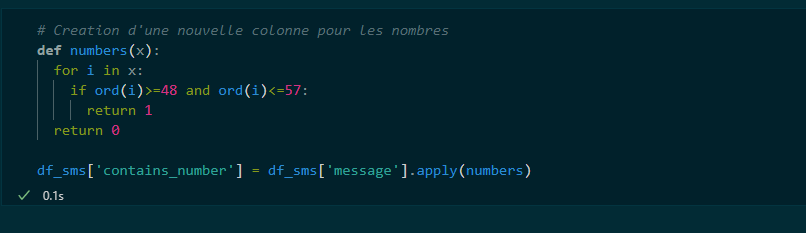
Sur le graphique ci-dessous nous pouvons visualiser la distribution de la nouvelle colonne *contains\_currency\_symbol* dans le jeu de donnée.



**Figure 10 : Graphique de la distribution de contain\_currency pour les chaque type de message**

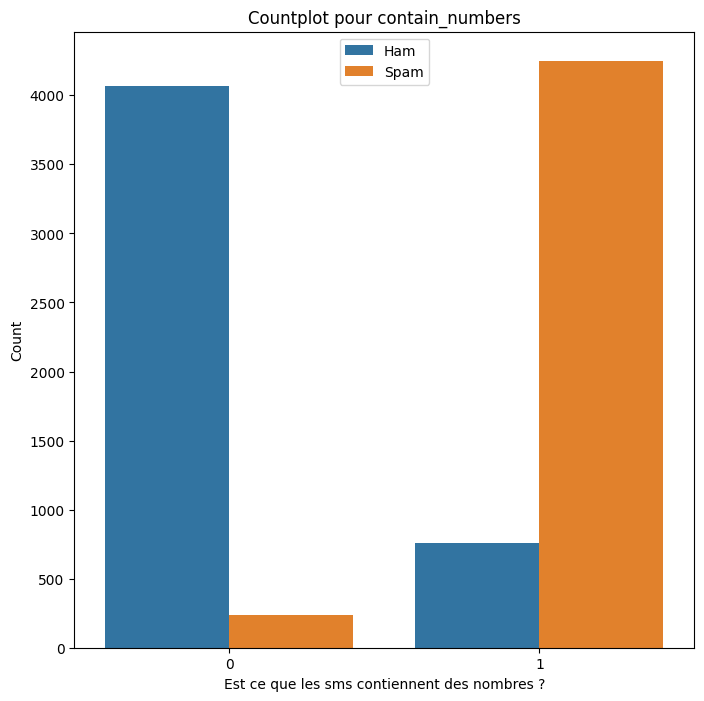
Près d'un tiers des messages de spam contiennent des symboles monétaires, et les symboles monétaires sont rarement utilisés dans les messages de Ham (normaux).

Nous allons appliquer le même procédé pour répertorier les nombres présentent dans les messages.



|  | **label** | **message** | **word\_count** | **contains\_currency\_symbol** | **contains\_number** |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | Go until jurong point, crazy.. Available only ... | 20 | 0 | 0 |
| 1 | 0 | Ok lar... Joking wif u oni... | 6 | 0 | 0 |
| 2 | 1 | Free entry in 2 a wkly comp to win FA Cup fina... | 28 | 0 | 1 |
| 3 | 0 | U dun say so early hor... U c already then say... | 11 | 0 | 0 |
| 4 | 0 | Nah I don't think he goes to usf, he lives aro... | 13 | 0 | 0 |

On note la présence de nombres dans le contenu de certains messages.



**Figure 11 : Graphique de la distribution de contain\_number pour les chaque type de message**

Il est évident que la plupart des messages de spam contiennent des chiffres et que la majorité des messages de Ham ne contiennent pas de chiffres.

Une fois la partie de l’exploration des données effectuée nous allons à présent procéder au nettoyage ou prétraitement des données dans la partie suivante.

## **Nettoyage et prétraitement des données**

Le prétraitement des données est un processus de préparation des données brutes et de leur adaptation à un modèle d'apprentissage automatique. C'est la première étape cruciale lors de la création d'un modèle d'apprentissage automatique.

Lors de la création d'un projet d'apprentissage automatique, nous ne rencontrons pas toujours des données propres et formatées. Et lors de toute opération avec des données, il est obligatoire de les nettoyer et de les mettre en forme.

Une donnée de la vraie vie contient généralement des bruits, des valeurs manquantes et peut-être dans un format inutilisable. Le prétraitement des données est une tâche requise pour nettoyer les données et les adapter à un modèle d'apprentissage automatique, ce qui augmente également la précision et l'efficacité d'un modèle d'apprentissage automatique.

Cette procédure se fera en différentes étapes. La première étape consiste en un nettoyage des données incorrectes, incomplètes ou manquantes. Il existe plusieurs manières de gérer ces problèmes.

Dans notre cas aucune données manquantes n’a été détecté raison pour laquelle nous allons passer cette étape.

Il peut arriver que des données souffrent d’un bruit parasite d’acquisition, auquel cas, elles ne pourront pas être correctement traitées par un ordinateur. Une manière de traiter ce problème est d’effectuer un *binning* des données (préalablement triées). Ces dernières sont séparées en groupe de même taille et chaque groupe est traité indépendamment. Au sein d’un même groupement, toutes les données peuvent être remplacées par leur moyenne, leur médiane ou par les valeurs extrêmes.

Une autre façon de traiter des données bruitées, et d’utiliser une régression ou un clustering, créant automatiquement des groupes de données et qui pourront nous permettre de détecter les *outliers* et de les supprimer de la base de donnée.

*Data Transformation* : Cette étape de prétraitement regroupe les changements effectués sur la structure même de la donnée. Ces transformations sont liées aux définitions mathématiques des algorithmes et à la manière dont ceux-ci traitent les données, de manière à optimiser les performances. Parmis ces techniques, nous pouvons citer par exemple :

Le lissage des données si elles sont bruitées.

L’agrégation de données venant de plusieurs sources différentes.

La discrétisation de variables continus (à l’aide du découpage en intervalles) qui permet d’abaisser le nombre de modalités d’un descripteur, et enfin.

La normalisation et la standardisation des données qui ramènent les données numériques à une échelle plus petite (par exemple entre -1 et 1), qui peuvent également centrer la moyenne et réduire la variance.

Dans notre projet, au regard des types de données dont nous disposons, nous aurons juste besoins de faire le nettoyage des messages et enlever tout autre caractère autre que des lettres.

Nous allons aussi utiliser certaines techniques du langage Naturel (NLP) pour le nettoyage.

### **Qu’est-ce que le NLP ?**

Le NLP pour Natural Language Processing ou Traitement du Langage Naturel est une discipline qui porte essentiellement sur la compréhension, la manipulation et la génération du langage naturel par les machines. Ainsi, le NLP est réellement à l’interface entre la science informatique et la linguistique. Il porte donc sur la capacité de la machine à interagir directement avec l’humain.

Le NLP est terme assez générique qui recouvre un champ d’application très vaste. Voici les applications les plus populaires :

***Traduction automatique* :** Le développement d’algorithmes de traduction automatique a réellement révolutionné la manière dont les textes sont traduits aujourd’hui. Des applications, telles que Google Translator, sont capables de traduire des textes entiers sans aucune intervention humaine.

Le langage naturel étant par nature ambigu et variable, ces applications ne reposent pas sur un travail de remplacement mot à mot, mais nécessitent une véritable analyse et modélisation de texte, connue sous le nom de Traduction automatique statistique (Statistical Machine Translation en anglais).

***Sentiment analysis* :** Aussi connue sous le nom de « Opinion Mining », l’analyse des sentiments consiste à identifier les informations subjectives d’un texte pour extraire l’opinion de l’auteur.

À titre exemple, lorsqu’une marque lance un nouveau produit, elle peut exploiter les commentaires recueillis sur les réseaux sociaux pour identifier le sentiment positif ou négatif globalement partagé par les clients.

De manière générale, l’analyse des sentiments permet de mesurer le niveau de satisfaction des clients vis-à-vis des produits ou services fournis par une entreprise ou un organisme. Elle peut même s’avérer bien plus efficace que des méthodes classiques comme les sondages.

En effet, si l’on rechigne souvent à passer du temps à compléter de longs questionnaires, une partie croissante des consommateurs partage aujourd’hui fréquemment leurs opinions sur les réseaux sociaux. Ainsi, la recherche de textes négatifs et l’identification des principales plaintes permettent d’améliorer les produits, d’adapter la publicité et de réduire le niveau d’insatisfaction des clients.

***Marketing*** : Les spécialistes du marketing utilisent également le NLP pour rechercher des personnes étant susceptible d’effectuer un achat.

Ils s’appuient pour cela sur le comportement des internautes sur les sites, les réseaux sociaux et les requêtes aux moteurs de recherche. C’est grâce à ce type d’analyse que Google génère un profit non négligeable en proposant la bonne publicité aux bons internautes. Chaque fois qu’un visiteur clique sur une annonce, l’annonceur reverse jusqu’à 50 dollars !

De manière plus générale, les méthodes de NLP peuvent être exploitées pour dresser un portrait riche et complet du marché existant, des clients, des problèmes, de la concurrence et du potentiel de croissance des nouveaux produits et services de l’entreprise.

***Chatbots*** : Les méthodes NLP sont au cœur du fonctionnement des Chatbots actuels. Bien que ces systèmes ne soient pas totalement parfaits, ils peuvent aujourd’hui facilement gérer des tâches standards telles renseigner des clients sur des produits ou services, répondre à leurs questions, etc. Ils sont utilisés par plusieurs canaux, dont l’Internet, les applications et les plateformes de messagerie. L’ouverture de la plateforme Facebook Messenger aux chatbots en 2016 a contribué à leur développement.

*Autres domaines d'application :*

***Classification de texte*** : cela consiste à attribuer un ensemble de catégories prédéfinies à un texte donné. Les classificateurs de texte peuvent être utilisés pour organiser, structurer et catégoriser un ensemble de textes.

***Reconnaissance de caractères*** : Cela permet d’extraire, à partir de la reconnaissance des caractères, les principales informations des reçus, des factures, des chèques, des documents de facturation légaux, etc.

***Correction automatique*** : la plupart des éditeurs de texte sont aujourd’hui muni d’un correcteur orthographique qui permet de vérifier si le texte contient des fautes d’orthographe.

***Résumé automatique*** : les méthodes NLP sont également utilisées pour produire des résumés courts, précis et fluides d’un document texte plus long.

Parmis les principales étapes nous pouvons citer :

***Nettoyage*** : Variable selon la source des données, cette phase consiste à réaliser des tâches telles que la suppression d’urls, d’emoji, etc.

Dans notre cas de figure nous avons procédé au nettoyage des caractères spéciaux, des symboles de devises, ainsi qu’à la conversion des toutes les lettres en minuscule.

**Tokenisation**, ou découpage du texte en plusieurs pièces appelés tokens.

**Stemming** : un même mot peut se retrouver sous différentes formes en fonction du genre (masculin féminin), du nombre (singulier, pluriel), la personne (moi, toi, eux…) etc. Le stemming désigne généralement le processus heuristique brut qui consiste à découper la fin des mots afin de ne conserver que la racine du mot.

**Lemmatisation** : cela consiste à réaliser la même tâche mais en utilisant un vocabulaire et une analyse fine de la construction des mots. La lemmatisation permet donc de supprimer uniquement les terminaisons inflexibles et donc à isoler la forme canonique du mot, connue sous le nom de lemme.

Autres opérations : suppression des chiffres, ponctuation, symboles et stopwords, passage en minuscule.

La capture suivante montre les opérations de nettoyage effectuées dans notre jeu de données.

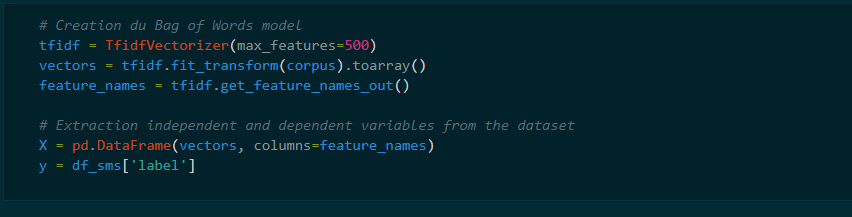


Afin de pouvoir appliquer les méthodes de Machine Learning aux problèmes relatifs au langage naturel, il est indispensable de transformer les données textuelles en données numériques.

Il existe plusieurs approches dont les principales sont les suivantes :

*Term-Frequency (TF)* : cette méthode consiste à compter le nombre d’occurrences des tokens présents dans le corpus pour chaque texte. Chaque texte est alors représenté par un vecteur d’occurrences. On parle généralement de **Bag-Of-Word**, ou sac de mots en français.

Nous avons effectué cette procédure au niveau de notre jeu de donnée comme l’atteste la capture de notre code suivante :



Toute ces techniques de nettoyage ont été effectuées à l’aide de librairies indispensable du NLP (Language Naturel) parmi lesquelles nous pouvons citer :

### **NLTK**

Le NLTK, ou Natural Language Toolkit, est une suite de bibliothèques logicielles et de programmes. Elle est conçue pour le traitement naturel symbolique et statistique du langage anglais en langage Python. C’est l’une des bibliothèques de traitement naturel du langage les plus puissantes.

Cette suite d’outils rassemble les algorithmes les plus communs du traitement naturel du langage comme le tokenizing, le part-of-speech tagging, le stemming, l’analyse de sentiment, la segmentation de topic ou la reconnaissance d’entité nommée.



### **Module RE**

Les expressions régulières (regex) sont une séquence de caractères qui définissent un modèle de recherche. Elles sont utilisées dans le cadre du NLP pour rechercher des modèles ou des structures spécifiques dans des données textuelles. Les expressions régulières sont très expressives et peuvent correspondre à de nombreux motifs, notamment des nombres, des dates, des adresses électroniques et des numéros de téléphone.

Une expression régulière est un moyen puissant de décrire des motifs complexes de manière concise et lisible. Elles constituent donc un puissant outil de prétraitement et de nettoyage des données, une étape essentielle dans de nombreuses applications NLP.

En NLP, les expressions régulières sont utilisées pour diverses tâches, telles que :

Le nettoyage et le prétraitement des données : Les expressions régulières peuvent nettoyer les données textuelles en supprimant les caractères indésirables, tels que les caractères spéciaux, la ponctuation et les chiffres, qui peuvent interférer avec les algorithmes et les modèles de NLP.

Reconnaissance des entités nommées (NER) : Les entités nommées, telles que les noms, les lieux et les organisations, jouent un rôle important dans de nombreuses applications NLP, telles que l'analyse des sentiments et l'extraction d'événements. Les expressions régulières peuvent extraire des entités nommées à partir de données textuelles, ce qui permet d'effectuer la reconnaissance d'entités nommées de manière plus efficace.

Normalisation du texte : Les expressions régulières peuvent être utilisées pour normaliser des données textuelles en les transformant dans un format standard, par exemple en convertissant tout le texte en minuscules ou en supprimant les mots vides. Cela permet d'effectuer des tâches NLP de manière plus efficace.

### **Scikit-learn**

Scikit-learn est une bibliothèque libre Python destinée à l'apprentissage automatique. Elle est développée par de nombreux contributeurs notamment dans le monde académique par des instituts français d'enseignement supérieur et de recherche comme Inria.

Elle propose dans son Framework de nombreuses bibliothèques d’algorithmes à implémenter, clé en main. Ces bibliothèques sont à disposition notamment des data scientists.

Elle comprend notamment des fonctions pour estimer des forêts aléatoires, des régressions logistiques, des algorithmes de classification, et les machines à vecteurs de support. Elle est conçue pour s'harmoniser avec d'autres bibliothèques libres Python, notamment NumPy et SciPy.



Nous avons aussi fait recours à certains modules comme le **WordNetLemmatizer** utilisé pour faire la lemmatisation des mots et le **TfidfVectorizer** qui permet de Convertir une collection de documents bruts en une matrice de caractéristiques TF-IDF**.**

La collecte et l’analyse de données représente une phase cruciale dans tout processus de conception d’un modèle d’apprentissage automatique (Machine Learning). C’est au niveau de cette phase que les opérations préliminaires s’effectuerons à savoir l’exploration et le prétraitement des données collectées afin de pouvoir mettre en place des modèles d’apprentissage fiables et efficaces.

# **Mise en place du modèle d’apprentissage automatique (ML)**

## **Les différents types de modèles d’apprentissage**

Le développement d’un modèle de Machine Learning repose sur quatre étapes principales.

La première étape consiste à sélectionner et à préparer un ensemble de données d’entraînement. Ces données seront utilisées pour nourrir le modèle de Machine Learning pour apprendre à résoudre le problème pour lequel il est conçu.

Les données peuvent être étiquetées, afin d’indiquer au modèle les caractéristiques qu’il devra identifier. Elles peuvent aussi être non étiquetées, et le modèle devra repérer et extraire les caractéristiques récurrentes de lui-même.

Dans les deux cas, les données doivent être soigneusement préparées, organisées et nettoyées. Dans le cas contraire, l’entraînement du modèle de Machine Learning risque d’être biaisé. Les résultats de ses futures prédictions seront directement impactés.

La deuxième étape consiste à sélectionner un algorithme à exécuter sur l’ensemble de données d’entraînement. Le type d’algorithme à utiliser dépend du type et du volume de données d’entraînement et du type de problème à résoudre.

La troisième étape est l’entraînement de l’algorithme. Il s’agit d’un processus itératif. Des variables sont exécutées à travers l’algorithme, et les résultats sont comparés avec ceux qu’il aurait dû produire. Les poids et le biais peuvent ensuite être ajustés pour accroître la précision du résultat.

On distingue trois techniques de Machine Learning : l’apprentissage supervisé, l’apprentissage non-supervisé, et l’apprentissage par renforcement.

### **L’apprentissage supervisé**

Dans le cas de l’apprentissage supervisé, le plus courant, les données sont étiquetées afin d’indiquer à la machine quelles patterns elle doit rechercher.

Le système s’entraîne sur un ensemble de données étiquetées, avec les informations qu’il est censé déterminer. Les données peuvent même être déjà classifiées de la manière dont le système est supposé le faire.

Cette méthode nécessite moins de données d’entraînement que les autres, et facilite le processus d’entraînement puisque les résultats du modèle peuvent être comparés avec les données déjà étiquetées. Cependant, l’étiquetage des données peut se révéler onéreux. Un modèle peut aussi être biaisé à cause des données d’entraînement, ce qui impactera ses performances par la suite lors du traitement de nouvelles données.

### **L’apprentissage non supervisé**

Dans le cas de l’apprentissage non supervisé, les données n’ont pas d’étiquettes. La machine se contente d’explorer les données à la recherche d’éventuelles patterns. Elle ingère de vastes quantités de données, et utilise des algorithmes pour en extraire des caractéristiques pertinentes requises pour étiqueter, trier et classifier les données en temps réel sans intervention humaine.

Plutôt que d’automatiser les décisions et les prédictions, cette approche permet d’identifier les patterns et les relations que les humains risquent de ne pas identifier dans les données. Cette technique n’est pas très populaire, car moins simple à appliquer. Elle est toutefois de plus en plus populaire dans le domaine de la cyber sécurité.

### **L’apprentissage par renforcement**

L’apprentissage par renforcement consiste à laisser un algorithme apprendre de ses erreurs pour atteindre un objectif. L’algorithme essayera de nombreuses approches différentes pour tenter d’atteindre son but.

En fonction de ses performances, il sera récompensé ou pénalisé pour l’inciter à poursuivre dans une voie ou à changer d’approche. Cette technique est notamment utilisée pour permettre à une IA de surpasser les humains dans les jeux.

Dans notre projet nous allons utiliser **l’apprentissage supervisé** notamment les méthodes de **classification.**

Comme définit au début de notre document, La **classification** est un processus d'apprentissage supervisé où un modèle est entraîné à prédire la catégorie ou la classe d'un nouvel exemple à partir d'un ensemble d'exemples d'apprentissage préalablement étiquetés. L'objectif de la classification est de généraliser à partir des exemples d'apprentissage pour classer correctement les nouvelles instances.

## **Choix du modèle de classification**

La classification offre plusieurs modèles permettant de faire un apprentissage automatique. Nous avons axé notre travail sur trois modèles sur lesquelles nous allons faire l’évaluation et l’entrainement afin d’en choisir un à déployer à la fin. Les trois modèles en question sont : *M****ultinomial*** ***Naive Bayes, Random Forest et Decision Tree***.

### **Multinomial Naive Bayes**

L'algorithme multinomial de Naive Bayes est une méthode d'apprentissage probabiliste principalement utilisée dans le traitement du langage naturel (NLP). L'algorithme est basé sur le théorème de Bayes et prédit l'étiquette d'un texte tel qu'un courriel ou un article de journal. Il calcule la probabilité de chaque étiquette pour un échantillon donné et donne ensuite l'étiquette ayant la probabilité la plus élevée comme résultat.

Le classificateur Naive Bayes est une collection de nombreux algorithmes qui partagent tous un principe commun, à savoir que chaque caractéristique classée n'est liée à aucune autre caractéristique. La présence ou l'absence d'une caractéristique n'affecte pas la présence ou l'absence d'une autre caractéristique.

**Avantages de l'algorithme**

L'algorithme de Naive Bayes présente les avantages suivants :

* Il est facile à mettre en œuvre puisqu'il suffit de calculer les probabilités.
* Vous pouvez utiliser cet algorithme sur des données continues et discrètes.
* Il est simple et peut être utilisé pour prédire des applications en temps réel.
* Il est très évolutif et peut facilement traiter de grands ensembles de données.

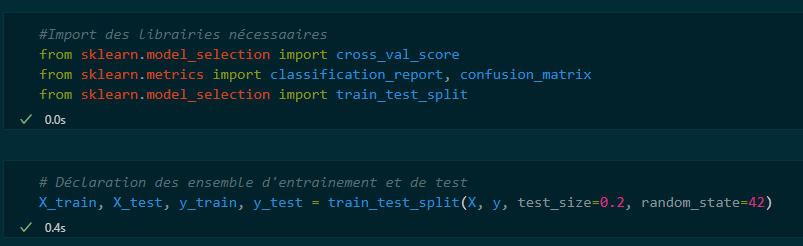
**Inconvénients** :

L'algorithme de Naive Bayes présente les inconvénients suivants :

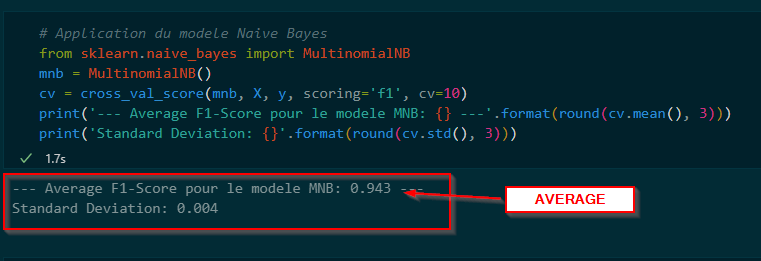
* La précision de prédiction de cet algorithme est inférieure à celle des autres algorithmes de probabilité.
* Il n'est pas adapté à la régression. L'algorithme de Naive Bayes n'est utilisé que pour la classification de données textuelles et ne peut pas être utilisé pour prédire des valeurs numériques.

Dans notre cas pratique nous avons procédé comme suit pour l’application du modèle Multinomial Naive Bayes.

Nous avons en premier lieu importer les bibliothèques nécessaires et faire la déclaration des ensemble d’entrainement et de test.

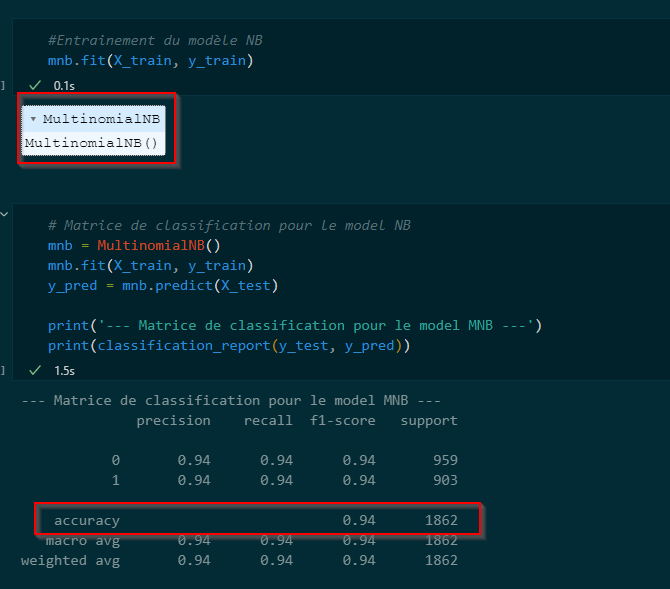


Ensuite l’application du modèle Naive avec le calcul du **F1-score** et de la **déviation** :



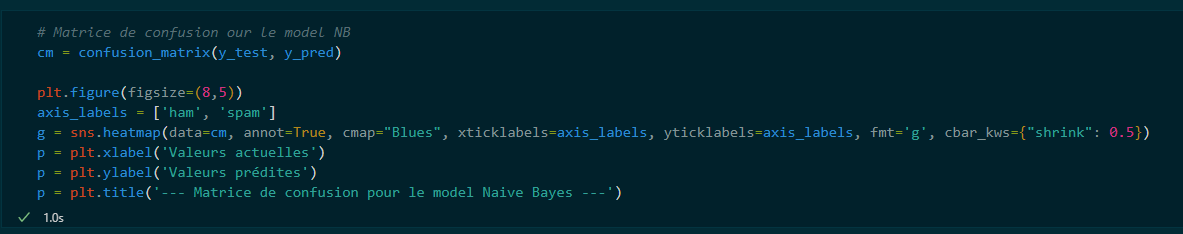
Nous avons ensuite procédé à l’entrainement ainsi qu’à l’affichage des matrices de classification et de confusion :

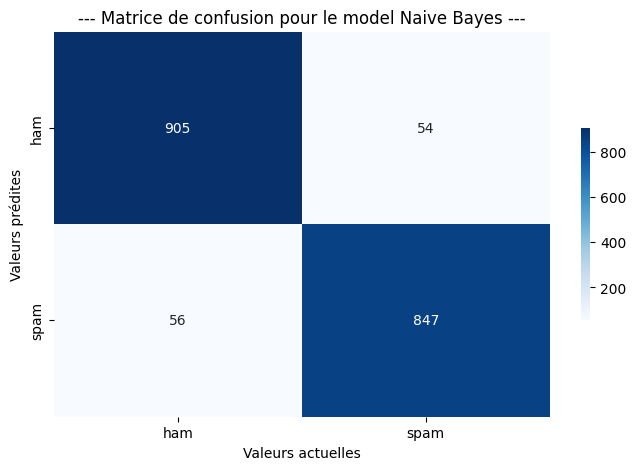
Matrice de classification :



**On note un accuracy de 0.94.**

La matrice de confusion :





**Figure 12 : Matrice de confusion du model Multinomial NB**

### **Decision Tree**

Les arbres de décision (DT) sont une méthode d'apprentissage supervisé non paramétrique utilisée pour la classification et la régression. L'objectif est de créer un modèle qui prédit la valeur d'une variable cible en apprenant des règles de décision simples déduites des caractéristiques des données. Un arbre peut être considéré comme une approximation constante par morceaux.

Les arbres de décision présentent les avantages suivants

* Ils sont simples à comprendre et à interpréter. Les arbres peuvent être visualisés.
* Nécessite peu de préparation des données. Les autres techniques nécessitent souvent une normalisation des données, la création de variables fictives et la suppression des valeurs manquantes. Certaines combinaisons d'arbres et d'algorithmes prennent en charge les valeurs manquantes.
* Le coût d'utilisation de l'arbre (c'est-à-dire la prédiction des données) est logarithmique dans le nombre de points de données utilisés pour former l'arbre.
* Capable de traiter à la fois des données numériques et catégorielles. Toutefois, l'implémentation de scikit-learn ne prend pas en charge les variables catégorielles pour le moment. Les autres techniques sont généralement spécialisées dans l'analyse d'ensembles de données ne comportant qu'un seul type de variable. Voir les algorithmes pour plus d'informations.
* Capable de traiter des problèmes à sorties multiples.

1. Utilise un modèle de boîte blanche. Si une situation donnée est observable dans un modèle, l'explication de la condition est facilement explicable par la logique booléenne. En revanche, dans un modèle à boîte noire (par exemple, dans un réseau neuronal artificiel), les résultats peuvent être plus difficiles à interpréter.
2. Possibilité de valider un modèle à l'aide de tests statistiques. Cela permet de rendre compte de la fiabilité du modèle.
3. Donne de bons résultats même si ses hypothèses sont quelque peu violées par le véritable modèle à partir duquel les données ont été générées.

Les inconvénients des arbres de décision sont les suivants :

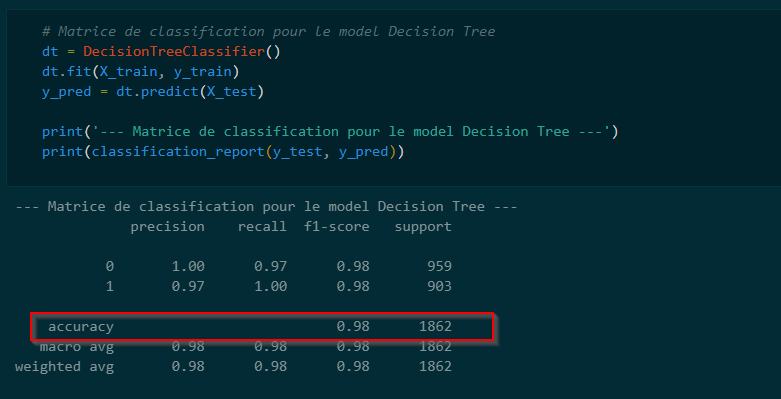
* Les apprenants d'arbres de décision peuvent créer des arbres trop complexes qui ne généralisent pas bien les données. C'est ce qu'on appelle l'adaptation excessive. Des mécanismes tels que l'élagage, la fixation du nombre minimum d'échantillons requis à un nœud feuille ou la fixation de la profondeur maximale de l'arbre sont nécessaires pour éviter ce problème.
* Les arbres de décision peuvent être instables, car de petites variations dans les données peuvent entraîner la génération d'un arbre complètement différent. Ce problème est atténué par l'utilisation d'arbres de décision au sein d'un ensemble.
* Les prédictions des arbres de décision ne sont ni lisses ni continues, mais des approximations constantes par morceaux, comme le montre la figure ci-dessus. Par conséquent, ils ne sont pas bons pour l'extrapolation.
* Le problème de l'apprentissage d'un arbre de décision optimal est connu pour être NP-complet sous plusieurs aspects de l'optimalité et même pour des concepts simples. Par conséquent, les algorithmes pratiques d'apprentissage d'arbres de décision sont basés sur des algorithmes heuristiques tels que l'algorithme gourmand, où des décisions localement optimales sont prises à chaque nœud. Ces algorithmes ne peuvent pas garantir l'obtention d'un arbre de décision globalement optimal. Ce problème peut être atténué en formant plusieurs arbres dans un apprenant d'ensemble, où les caractéristiques et les échantillons sont échantillonnés de manière aléatoire avec remplacement.
* Certains concepts sont difficiles à apprendre parce que les arbres de décision ne les expriment pas facilement, comme les problèmes de XOR, de parité ou de multiplexage.
* Les apprenants d'arbres de décision créent des arbres biaisés si certaines classes dominent. Il est donc recommandé d'équilibrer.

Application du modèle de Decision Tree sur notre cas pratique :

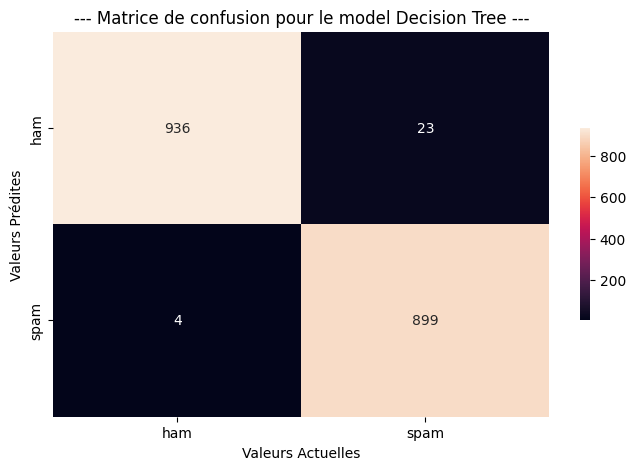


On note un **accuracy de 0.98** pour le modele Decision Tree**.**

Matrice de classification :



Matrice de confusion :



**Figure 13 : Matrice de confusion du model Decision Tree**

### **Random Forest**

Une Random Forest (ou Forêt d’arbres de décision en français) est une technique de Machine Learning très populaire auprès des Data Scientists et pour cause : elle présente de nombreux avantages comparé aux autres algorithmes de data.

C’est une technique facile à interpréter, stable, qui présente en général de bonnes accuracy et qui peut être utilisée pour des tâches de régression ou de classification. Elle couvre donc une grande partie des problèmes de Machine Learning.

Dans Random Forest il y a d’abord le mot « Forest » (ou forêt en français). On comprend donc que cet algorithme va reposer sur des arbres que l’on appelle arbre de décision ou arbre décisionnel.

Les principaux concepts associés au Random Forest :

*Ensemble d'arbres de décision* : Le Random Forest est construit en combinant plusieurs arbres de décision individuels. Chaque arbre de décision est formé sur un sous-ensemble aléatoire des données d'entraînement et utilise un sous-ensemble aléatoire des caractéristiques à chaque nœud de décision.

*Bagging (Bootstrap Aggregating)* : Le processus de construction du Random Forest repose sur la technique de Bagging, qui consiste à former chaque arbre sur un échantillon bootstrap de l'ensemble d'entraînement. Cela signifie qu'un échantillon est tiré aléatoirement avec remplacement à partir des données d'entraînement pour former chaque arbre.

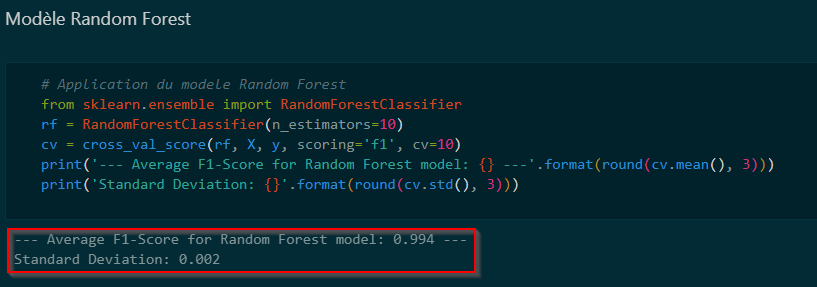
*Sous-ensembles aléatoires de caractéristiques* : Lors de la construction de chaque arbre, un sous-ensemble aléatoire de caractéristiques est utilisé à chaque nœud de décision. Cela augmente la diversité des arbres, ce qui contribue à améliorer la généralisation du modèle.

*Vote majoritaire* : Pour la classification, le Random Forest utilise un vote majoritaire pour attribuer une classe à une observation. Chaque arbre vote pour une classe, et la classe ayant le plus de votes est attribuée à l'observation. Pour la régression, la prédiction est souvent la moyenne des prédictions de tous les arbres.

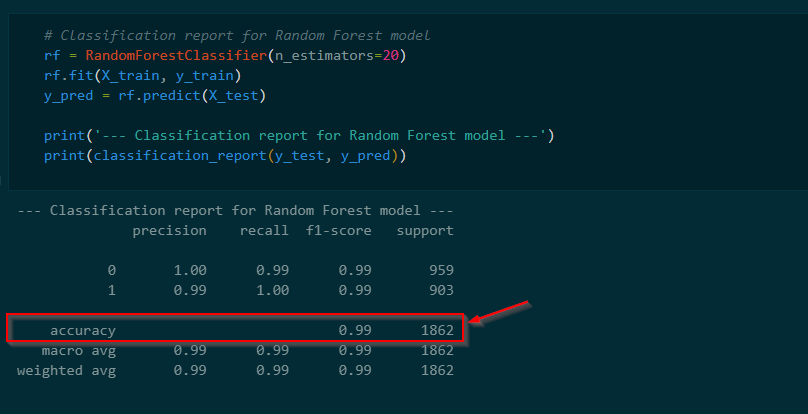
*Robustesse et Réduction de l'overfitting* : En raison de la nature aléatoire de la construction des arbres et de l'utilisation de plusieurs arbres, le Random Forest est généralement robuste aux données bruitées et a une meilleure capacité à généraliser aux données invisibles par rapport à un seul arbre de décision.

*Importance des Caractéristiques* : Le Random Forest fournit une mesure de l'importance de chaque caractéristique dans la prise de décision. Cela peut être utile pour comprendre quelles caractéristiques contribuent le plus à la prédiction.

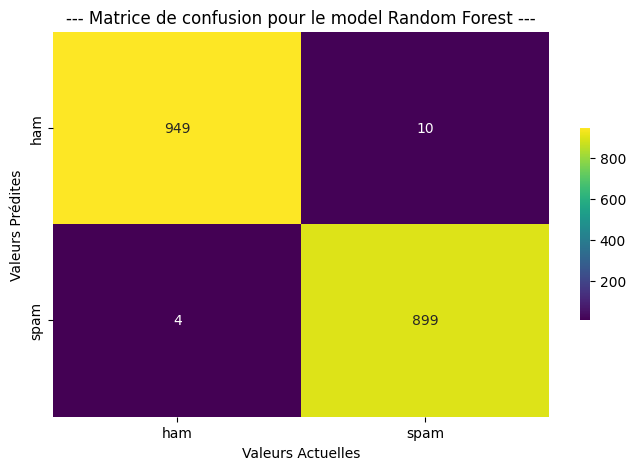
Application du modèle de Random Forest dans notre cas pratique :



Matrice de classification :



Matrice de confusion :



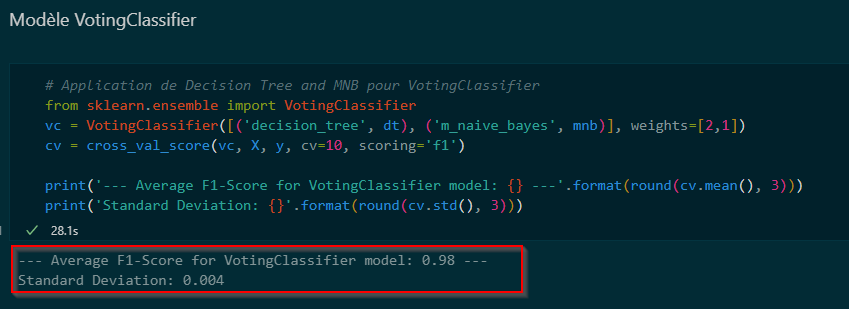
**Figure 14 : Matrice de confusion du model Random Forest**

A la suite de l’application de ces trois modèles, nous avons remarqué que le modèle Random Forest demeure le plus performant avec un accuracy de 0,99. Cependant nous allons faire appel à un quatrième modèle appelé **VotingClassifier**.

### **VotingClassifier**

Un classificateur de vote est un modèle d'apprentissage automatique qui s'entraîne sur un ensemble de nombreux modèles et prédit un résultat (classe) sur la base de la probabilité la plus élevée de choisir la classe comme résultat. Il agrège simplement les résultats de chaque classificateur passé dans le classificateur de vote et prédit la classe de sortie sur la base de la plus grande majorité des votes. L'idée est qu'au lieu de créer des modèles dédiés distincts et de déterminer la précision pour chacun d'entre eux, nous créons un modèle unique qui entraîne ces modèles et prédit la sortie sur la base de leur majorité de vote combinée pour chaque classe de sortie.

Dans notre cas pratique nous avons intégré les deux modèles les moins performants à savoir le Naive Bayes et le Decision Tree au modèle VotingClassifier.



**Average 0.98**

L'algorithme de vote n'a pas été plus performant que l'algorithme **Random Forest**, c'est pourquoi l'algorithme Random Forest est sélectionné pour prédire les résultats de cet énoncé de problème.

Une fois le modèle choisit, nous allons à présent procéder à la prédiction.

## **Prédiction du modèle choisit : Random Forest**

L'étape de la prédiction en Machine Learning consiste à utiliser un modèle entraîné pour faire des prédictions sur de nouvelles données.

Dans notre cas pratique nous avons défini une fonction dans laquelle la prédiction a été effectué.